



**HAL**  
open science

# Optimal Trajectories Generation in Robotic Fiber Placement Systems

Jiuchun Gao, Anatol Pashkevich, Stéphane Caro

► **To cite this version:**

Jiuchun Gao, Anatol Pashkevich, Stéphane Caro. Optimal Trajectories Generation in Robotic Fiber Placement Systems. The 4th International Conference on Manufacturing and Industrial Technologies, May 2017, Lisbon, Portugal. 10.1088/1757-899X/212/1/012009 . hal-01757920

**HAL Id: hal-01757920**

**<https://hal.science/hal-01757920v1>**

Submitted on 4 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Optimal Trajectories Generation in Robotic Fiber Placement Systems

Jiuchun Gao<sup>1,\*</sup>, Anatol Pashkevich<sup>1</sup>, Stéphane Caro<sup>2</sup>

<sup>1</sup>IMT Atlantique Engineering School, Laboratory LS2N, France

<sup>2</sup>National Center for Scientific Research CNRS, Laboratory LS2N, France

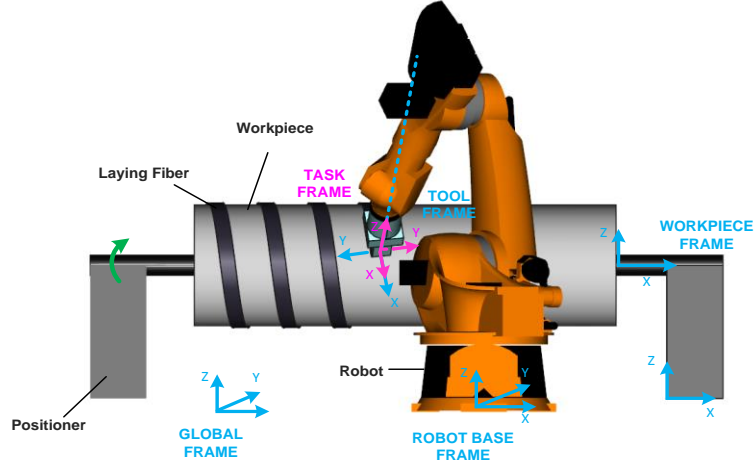
\*Email: [jiuchun.gao@ls2n.fr](mailto:jiuchun.gao@ls2n.fr)

**Abstract.** The paper proposes a methodology for optimal trajectories generation in robotic fiber placement systems. A strategy to tune the parameters of the optimization algorithm at hand is also introduced. The presented technique transforms the original continuous problem into a discrete one where the time-optimal motions are generated by using dynamic programming. The developed strategy for the optimization algorithm tuning allows essentially reducing the computing time and obtaining trajectories satisfying industrial constraints. Feasibilities and advantages of the proposed methodology are confirmed by an application example.

## 1. Introduction

Robotic fiber placement technology has been increasingly implemented recently in aerospace and automotive industries for fabricating complex composite parts [1, 2]. It is a specific technique that uses robotic workcell to place the heated fiber tows on the workpiece surface [3]. Corresponding robotic systems usually include a 6-axis industrial robot and a one-axis positioner (see Figure 1), which are kinematically redundant and provides the user with some freedom in terms of optimization of robot and positioner motions.

To deal with the robotic system redundancy, a common technique based on the pseudo-inverse of kinematic Jacobian is usually applied. However, as follows from relevant studies, this standard approach does not satisfy the real-life industrial requirements of the fiber placement [4, 5]. In literature, there is also an alternative technique (that deals with multi-goal tasks) that is based on conversion of the original continuous problem to the combinatorial one [6, 7], but it only generates trajectories for *point-to-point* motions, e.g. for spot welding applications. A slightly different method was introduced in [8-10], and it was successfully applied to laser-cutting and arc-welding processes where the tool speed was assumed to be constant (which is not valid in the considered problem). Another approach has been proposed in [11], where the authors concentrated on the tool path smoothing in Cartesian space in order to decrease the manufacturing time in fiber placement applications. For the considered process, where the tool speed variations are allowed (in certain degree), a discrete optimization based methodology was proposed in our previous work [12]. It allows the user to convert the original continuous problem to the combinatorial one taking into account particularities of the fiber placement technology and to generate time-optimal trajectories for both the robot and the positioner. Nevertheless, there are still a number of open questions related to selection of the optimization algorithm parameters (i.e., its “tuning”) that are addressed in this paper and targeted to the improvement of the algorithm efficiency and the reduction of the computing time.



**Figure 1.** Typical robotic fiber placement system (6-axis robot and one-axis positioner).

## 2. Robotic system model

In practice, the procedure of off-line programming for robotic fiber placement is implemented in the following way. The fiber placement path is firstly generated and discretized in CAM system. Further, the obtained set of task points is transformed into the task graph that describes all probable configurations of the robot and the positioner joints. The motion generator module finds the optimal trajectories that are presented as the “best” path on the graph. Finally, the obtained motions are converted into the robotic system program by the post processor. The core for the programming of this task is a set of optimization routines addressed in this paper.

To describe the fiber placement task, let us present it as a set of discrete task frames  $F_{task}^{(i)}, i = 1, 2, \dots, n$ , in such a way that the X-axis is directed along the path direction and Z-axis is normal to the workpiece surface pointing outside of it (see Figure 1). Using these notations, the task locations can be described by  $4 \times 4$  homogenous transformation matrices and the considered task is formalized as follows:

$${}^w\mathbf{T}_{task}^{(1)} \rightarrow {}^w\mathbf{T}_{task}^{(2)} \rightarrow \dots \rightarrow {}^w\mathbf{T}_{task}^{(i)} \rightarrow \dots \rightarrow {}^w\mathbf{T}_{task}^{(n)} \quad (1)$$

where all vectors of positions and orientations are expressed with respect to the workpiece frame (see superscript “w”). To execute the given fiber placement task, the robot tool must visit the frames defined by (1) as fast as possible.

The considered robotic system, shown in Figure 1, is composed of an industrial robot and an actuated positioner. Their spatial configurations can be described by the joint coordinates  $\mathbf{q}_R$  and  $q_P$  respectively. The task frames can be presented in two ways using the robot and positioner kinematics that are expressed as  $g_R(\mathbf{q}_R)$  and  $g_P(q_P)$ , respectively. To obtain the kinematic model of the whole system that is expressed as a closed loop containing the robot, the workpiece and the positioner, a global frame  $F_0$  is selected. Then, the tool frame  $F_{tool}$ , and task frame  $F_{task}^{(i)}$  can be aligned in such a way that: (i) the origins of the two frames coincide; (ii) Z-axes are opposite; (iii) X-axes have the same direction. Due to the foregoing closed-loop, two paths can be followed to express the transformation matrices from the global frame to the task frames, namely,

$${}^0\mathbf{T}_{Rbase} \cdot g_R(\mathbf{q}_R) \cdot {}^{Tool}\mathbf{T}_{task} = {}^0\mathbf{T}_{Pbase} \cdot g_P(q_P) \cdot {}^w\mathbf{T}_{task}^{(i)}; i = 1, 2, \dots, n \quad (2)$$

Equation (2) does not lead to a unique solution for  $\mathbf{q}_R$  and  $q_P$  as the robotic system, i.e., robot and positioner, is kinematically redundant. Therefore, the optimum robot and positioner configurations can be searched based on specific criteria.

## 3. Algorithm for trajectories generation

To take advantage of the kinematic redundancy, it is reasonable to partition the desired motion between the robot and the positioner ensuring that the technology tool executes the given task with smooth motion as fast as possible.

To present the problem in a formal way, let us define the functions  $\mathbf{q}_R(t)$  and  $q_p(t)$  describing the robot and positioner motion as a function of time  $t \in [0, T]$ . Additionally, a sequence of time instants  $\{t_1, t_2, \dots, t_n\}$  corresponds to the cases where the tool visits the locations defined by (1), and  $t_1 = 0, t_n = T$ . As a result, the problem at hand is formulated as an optimization problem aiming at minimizing the robot processing time

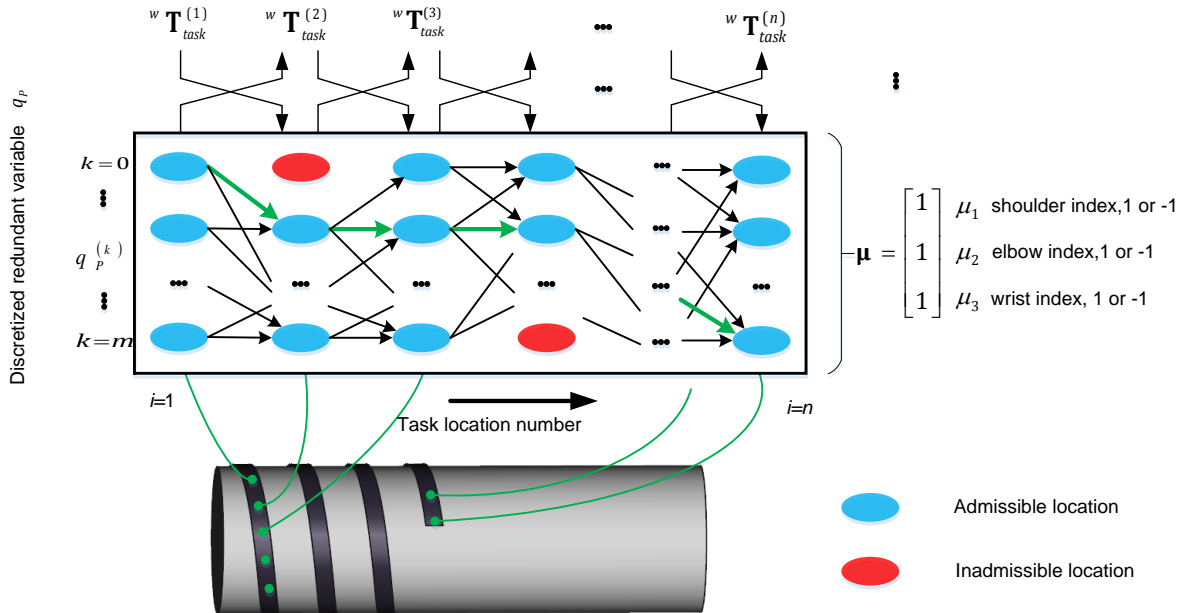
$$T \rightarrow \min_{\mathbf{q}_R(t), q_p(t)} \quad (3)$$

This problem is subjected to the equality constraints  ${}^0\mathbf{T}_{Rbase} \cdot \mathbf{g}_R(\mathbf{q}_R(t_i)) \cdot {}^{Tool}\mathbf{T}_{task} = {}^0\mathbf{T}_{Pbase} \cdot \mathbf{g}_P(q_p(t_i)) \cdot {}^w\mathbf{T}_{task}^{(i)}$  defined in (2) and some inequality constraints associated to the capacities of the robot/positioner actuators that are defined by upper bounds of the joint velocities and accelerations. Besides, the collision constraints verifying the intersections between the system components are also taken into account.

For this considered problem aiming at finding desired continuous function of  $\mathbf{q}_R(t)$  and  $q_p(t)$ , there is no standard approach that can be applied to straightforwardly. The main difficulty here is that the equality constraints are written for the unknown time instants  $\{t_1, t_2, \dots, t_n\}$ . Besides, this problem is nonlinear and includes a redundant variable. For these reasons, this paper presents a combinatorial optimization based methodology to generate the desired trajectories.

For the considered robotic system, there is one redundant variable with respect to the given task. It is convenient here to treat  $q_p$  as the redundant one since it allows us to use the kinematic models of the robot and the positioner independently and to consider the previous equality constraints.

To present the problem in a discrete way, the allowable domain of  $q_p \in [q_p^{\min}, q_p^{\max}]$  is sampled with the step  $\Delta q_p$  as  $q_p^{(k)} = q_p^{\min} + \Delta q_p \cdot k; k = 0, 1, \dots, m$ , where  $m = (q_p^{\max} - q_p^{\min}) / \Delta q_p$ . Then, applying sequentially the positioner direct kinematics and the robot inverse kinematics, a set of possible configuration states for the robotic system can be obtained as  $\mathbf{q}_R^{(k)}(t_i) = \mathbf{g}_R^{-1}({}^{Rbase}\mathbf{T}_0 \cdot {}^0\mathbf{T}_{Pbase} \cdot \mathbf{g}_P(q_p^{(k)}(t_i)) \cdot {}^w\mathbf{T}_{task}^{(i)} \cdot {}^{task}\mathbf{T}_{tool}^{-1}, \boldsymbol{\mu}); i = 1, 2, \dots, n$ , where  $\boldsymbol{\mu}$  is a configuration index vector corresponding to the robot posture. Therefore, for  $i^{\text{th}}$  task location, a set of candidate configuration states can be obtained, i.e.,  ${}^w\mathbf{T}_{task}^{(i)} \rightarrow \mathbf{L}_{task}^{(k;i)}, \forall k$ , where  $\mathbf{L}_{task}^{(k;i)} = (\mathbf{q}_R^{(k)}(t_i), q_p^{(k)}(t_i))$ .



**Figure 2.** Graph-based representation of the discrete search space

After presenting  ${}^w\mathbf{T}_{task}^{(i)}$  in joint space as above, the original task can be converted into the directed graph shown in Figure 2. It should be noted that some of the configuration cells should be excluded because of violation of the collision constraints or the actuator joint limits. These cases are denoted as “inadmissible” in Figure 2, and are not connected to any neighbor. Here, the allowable connection

between the graph nodes are limited to the subsequent configuration states  $\mathbf{L}_{task}^{(k',i)} \rightarrow \mathbf{L}_{task}^{(k,i)}$ , and the edge weights correspond to the minimum robot processing time restricted by the maximum velocities and accelerations of the robot and the positioner.

Using the discrete search space above, the considered problem is transformed to the classic shortest path searching and the desired solution can be represented as the sequence  $\{\mathbf{L}_{task}^{(k_1,1)}\} \rightarrow \{\mathbf{L}_{task}^{(k_2,2)}\} \rightarrow \dots \rightarrow \{\mathbf{L}_{task}^{(k_n,n)}\}$ . The objective function (robot processing time) can be presented as the sum of the edge weights

$$T = \sum_{i=1}^{n-1} dist(\mathbf{L}_{task}^{(k_i,i)}, \mathbf{L}_{task}^{(k_{i+1},i+1)}) \quad (4)$$

that depends on indices  $k_1, k_2, \dots, k_n$ , where  $dist(\mathbf{L}_{task}^{(k_i,i)}, \mathbf{L}_{task}^{(k_{i+1},i+1)}) = \max(|q_{j,i}^{(k_i)} - q_{j,i+1}^{(k_{i+1})}| / \dot{q}_j^{\max}); j=0,1,\dots,6$ . It should be mentioned that the above expression takes into account the velocity constraints automatically and the acceleration constraints should be considered by means of the following formula:

$$2|\Delta t_i (q_{j,i+1}^{(k_{i+1})} - q_{j,i}^{(k_i)}) - \Delta t_{i+1} (q_{j,i}^{(k_i)} - q_{j,i-1}^{(k_{i-1})})| / \Delta t_{i+1} \Delta t_i (\Delta t_{i+1} + \Delta t_i) \leq \ddot{q}_j^{\max} \quad (5)$$

where  $\Delta t_{i+1} = dist(\mathbf{L}_{task}^{(k_i,i)}, \mathbf{L}_{task}^{(k_{i+1},i+1)})$  and  $\Delta t_i = dist(\mathbf{L}_{task}^{(k_i,i)}, \mathbf{L}_{task}^{(k_{i-1},i-1)})$ .

By discretizing the search space, the original problem is converted to a combinatorial one, which can be solved by using conventional way, e.g. Dijkstra. However, this straightforward approach is extremely time-consuming and can be hardly accepted for industrial applications. For example, it takes over 20 hours to find a desired solution in a relatively simple case (two-axis robot and one-axis positioner), where the search space is built for 100 task points and the discretization step  $1^\circ$  (processor Intel® i5 2.67 GHz) [12]. Besides, known methods are not able to take into account the acceleration constraints that are necessary here. For these reasons, a problem-oriented algorithm taking into account the particularities of the graph based search space is proposed in this paper.

The developed algorithm is based on the dynamic programming principle, aiming at finding the shortest path from  $\{\mathbf{L}_{task}^{(k_1,1)}, \forall k_1\}$  to the current  $\{\mathbf{L}_{task}^{(k_i,i)}, \forall k_i\}$ . The length of this shortest path is denoted as  $d_{k,i}$ . Then, the shortest path for the locations corresponding to the next  $\{\mathbf{L}_{task}^{(k_{i+1},i+1)}, \forall k_{i+1}\}$  can be obtained by combining the optimal solutions for the previous column  $\{\mathbf{L}_{task}^{(k',i)}, \forall k'\}$  and the distances between the task locations with the indices  $i$  and  $i+1$ ,

$$d_{k,i+1} = \min_k \{d_{k',i} + dist(\mathbf{L}_{task}^{(k',i)}, \mathbf{L}_{task}^{(k,i+1)})\}; \quad (4)$$

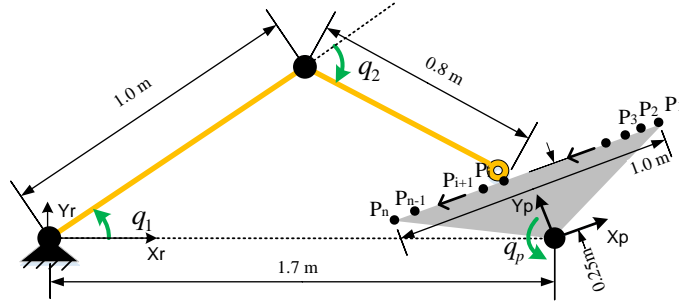
This formula is applied sequentially from the second column of the task graph to the last one, and the desired optimal path can be obtained after selection of the minimum length  $d_{k,i+1}$  corresponding to the final column. Therefore, the desired path is described by the recorded indices  $\{k_1, k_2, \dots, k_n\}$ . This proposed algorithm is rather time-efficient since it takes about 30 seconds [12] to find the optimal solution for the above mentioned example.

#### 4. Tuning of trajectories generation algorithm

For the proposed methodology, the discretization step for the redundant variable is a key parameter, which has a big influence on the algorithm efficiency. An unsuitable discretization step may lead either to a bad solution or high computational time. For this reason, a new strategy for the determination of the discretization step is proposed thereafter to tune the optimization algorithm.

##### 4.1. Influence of the discretization step

Let us consider a simple case study that deals with a three-axis planar robotic system executing a straight-line-task (see Figure 3). For this problem, the fiber placement path is uniformly discretized into 40 segments. Relevant optimization results are presented in Table 1. It is clear that here (as well as in other cases) smaller discretization step should provide better results but there exists a reasonable lower bound related to an acceptable computing time.



**Figure 3.** Three-axis planar robotic system and straight-line-task

**Table 1.** Optimization results and computing time for different discretization steps

	$\Delta q_p = 2^\circ$	$\Delta q_p = 1^\circ$	$\Delta q_p = 0.75^\circ$	$\Delta q_p = 0.5^\circ$	$\Delta q_p = 0.25^\circ$	$\Delta q_p = 0.1^\circ$
<b>Robot processing time</b> (without acc-constraint)	T=1.90s (~38s comp.)	T=1.84s (~2m comp.)	T=1.54s (~4m comp.)	T=1.30s (~9m comp.)	T=1.29s (~47m comp.)	T=1.29s (~5h comp.)
<b>Robot processing time</b> (with acc-constraint)	T=1.90s (~67s comp.)	T=2.11s (~4m comp.)	T=1.60s (~8m comp.)	T=1.30s (~17m comp.)	T=1.29s (~1.2h comp.)	T=1.29s (~9h comp.)

To estimate the reasonable discretization step for the considered fiber placement problem, let us analyze Table 1 in more details. From Table 1, the discretization steps  $\Delta q_p \in \{2^\circ, 1^\circ, 0.75^\circ\}$  are not acceptable because they lead to a robot processing time 20-50% higher than the optimal one. Moreover, in the case of  $\Delta q_p = 2^\circ$ , the optimization algorithm generates a bad solution that does not take advantage of the positioner motion capabilities (*positioner is locked,  $q_p = const$ ,  $\mathbf{q}_R = var$* ). The reason for this phenomenon is that the discretization step here is so large that the positioner step-time is always higher than the robot moving time between the subsequent task points.

Another interesting phenomenon can be observed for slightly smaller discretization steps, where the algorithm may produce non-smooth intermittent rotation of the positioner (*start-stop motion:  $q_p = const, \dots, (q_p + \Delta q_p), \dots, q_p = const$* ). For example, in the case of  $\Delta q_p = 1^\circ$  (without acceleration constraints), the optimization algorithm generates solution that includes only several steps where the positioner is not locked.

In addition, it is noteworthy that in the case of with acceleration constraints, the discretization step reduction from  $2^\circ$  to  $1^\circ$  leads to even worse solution, where the robot processing time is about 10% higher. This phenomenon can be explained by heuristic integration of the acceleration constraints into the optimization algorithm, which may slightly violate the dynamic programming principle. Nevertheless, further reduction of  $\Delta q_p$  allows to restore the expected algorithm behavior. Hence, to apply the developed technique in practice, users need some simple “rules of thumb” that allows setting an initial value of  $\Delta q_p$ . Then, the optimization algorithm can be applied several times (sequentially decreasing  $\Delta q_p$ ) until the objective function convergence. To reduce computing time in the case of small  $\Delta q_p$ , some local optimization techniques have been also developed by the authors.

#### 4.2. Initial tuning of the optimization algorithm

To find a reasonable initial value of the discretization step, let us investigate in details robot and positioner motions between two sequential task locations. It is clear that for smooth positioner motions, it is required that corresponding increments of the coordinate  $q_p$  should include at least one discretization step  $\Delta q_p$ .

To find the maximum value of  $\Delta q_p$ , let us denote  $\Delta\theta$  as the increment of  $q_p$  for the movement between two adjacent task locations ( $P_i$ - $P_{i+1}$ ) and  $\Delta s$  as the length of the path segment. It is clear that  $\Delta s$  can be also treated as the arc length between  $P_i$  and  $P_{i+1}$  around the positioner joint axis. Let us assume that the distance from a path point to the rotational axis is  $r$ , and  $r_{max}$  represents the furthest task location with respect to the positioner axis. To avoid undesired intermittent positioner rotations, the following constraint  $\Delta s > r_{max} \cdot \Delta\theta$  should be verified, since the positioner velocity is usually smaller than the velocity of the robot. The latter inequality can be rewritten in terms of the robot/positioner motion

time as  $(\Delta s - r_{\max} \cdot \Delta \theta) / v_R^{\max} = \Delta \theta / \dot{q}_P^{\max}$ , which can be further  $\Delta q_P \leq \Delta \theta$  where number of the positioner steps is no less one. Hence, the initial value of  $\Delta q_P$  should be at least equal to  $\dot{q}_P^{\max} \cdot \Delta s / (v_R^{\max} + \dot{q}_P^{\max} \cdot r_{\max})$  in order to provide acceptable motions of the robot and positioner. For instance, for the previous case study, this expression gives the discretization step about  $0.5^\circ$  that allows to generate trajectories that are very close to the optimal ones, namely, the robot processing time is only 1% higher than the minimum value.

## 5. Conclusions

This paper contributes to optimization of robot/positioner motions in redundant robotic systems for the fiber placement process. It proposes a new strategy for the optimization algorithm tuning. The developed technique converts the continuous optimization into a combinatorial one where dynamic programming is applied to find time-optimal motions. The proposed strategy of the optimization algorithm tuning allows essentially decreasing the computing time and generating desired motions satisfying industrial constraints. Feasibilities and advantages of the presented technique are confirmed by a case study. Future research will focus on application of those results in real-life industrial environments.

## Acknowledgments

This work has been supported by the China Scholarship Council (Grant N° 201404490018). The authors also acknowledge CETIM for the motivation of this research work.

## References

- [1] D. Gay, Composite materials: design and applications, CRC press, 2014.
- [2] C. Gallet-Hamlyn, Multiple-use robots for composite part manufacturing, JEC composites, (2011) 28-30.
- [3] S.T. Peters, Handbook of composites, Springer Science & Business Media, 2013.
- [4] K. Kazerounian, A. Nedungadi, Redundancy resolution of serial manipulators based on robot dynamics, Mechanism and machine theory, 23 (1988) 295-303.
- [5] S.R. Buss, Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods, IEEE Journal of Robotics and Automation, 17 (2004) 16.
- [6] L.B. Gueta, R. Chiba, J. Ota, T. Ueyama, T. Arai, Coordinated motion control of a robot arm and a positioning table with arrangement of multiple goals, Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE, 2008, pp. 2252-2258.
- [7] L.B. Gueta, R. Chiba, T. Arai, T. Ueyama, J. Ota, Hybrid design for multiple-goal task realization of robot arm with rotating table, Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 1279-1284.
- [8] A. Dolgui, A. Pashkevich, Manipulator motion planning for high-speed robotic laser cutting, International Journal of Production Research, 47 (2009) 5691-5715.
- [9] A.P. Pashkevich, A.B. Dolgui, O.A. Chumakov, Multiobjective optimization of robot motion for laser cutting applications, International Journal of Computer Integrated Manufacturing, 17 (2004) 171-183.
- [10] B. Zhou, X. Zhang, Z. Meng, X. Dai, Off-line programming system of industrial robot for spraying manufacturing optimization, Proceedings of the 33rd Chinese Control Conference, 2014.
- [11] P. Debout, H. Chanal, E. Duc, Tool path smoothing of a redundant machine: Application to Automated Fiber Placement, Computer-Aided Design, 43 (2011) 122-132.
- [12] J. Gao, A. Pashkevich, S. Caro, Manipulator Motion Planning in Redundant Robotic System for Fiber Placement Process, in: P. Wenger, P. Flores (Eds.) New Trends in Mechanism and Machine Science: Theory and Industrial Applications, Springer International Publishing, Cham, 2017, pp. 243-252.